

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

Claim 1 (Currently Amended): A method comprising:
establishing a secure connection between a network router and a client;
initiating a command line interface (CLI) process on the network router;
receiving from the client a CLI command; and
in response to the CLI command, accepting commands encoded in accordance with an extensible markup language, wherein accepting commands comprises replacing the CLI process with a management server process in response to the CLI command, wherein the management server process provides an extensible markup language-based application programming interface (API) to the client.

Claim 2 (Canceled).

Claim 3 (Previously Presented): The method of claim 1, further comprising:

receiving from the client configuration requests and operational requests encoded with extensible markup language tags;

accessing a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router, the software modules including a chassis software module that defines an inventory of components in the network router chassis, a device configuration software module that defines a physical configuration of the network router, and a routing protocol module that administers protocols supported by the network router;

parsing the configuration requests and the operational requests;

accessing the corresponding configuration and operational information associated with the chassis software module, the device configuration software module, and the routing control software module according to the network management interface schema; and

emitting replies encoded with extensible markup language tags according to the network management interface schema.

Claim 4 (Original): The method of claim 3, wherein the tags include chassis tags that define output for the chassis software module, device configuration tags that define output for the routing protocol software module.

Claim 5 (Original): The method of claim 3, further comprising receiving at least some of the replies encoded with extensible markup language tags from one or more of the software modules.

Claim 6 (Original): The method of claim 3, further comprising encoding at least some of the replies with extensible markup language tags based on non-encoded replies received from one or more of the software modules.

Claim 7 (Original): The method of claim 6, wherein the non-encoded replies include ASCII replies, the method further comprising encoding the ASCII replies in extensible markup language tags to form the encoded replies.

Claim 8 (Original): The method of claim 3, wherein the extensible markup language tags are defined by one or more data type of definition (DTD) files.

Claim 9 (Original): The method of claim 3, wherein the extensible markup language tags are defined by one or more XML Schema Language files.

Claim 10 (Original): The method of claim 3, wherein the configuration requests includes requests to change an existing configuration of the network router, the method further comprising interacting with one or more of the software modules to effect the requested change.

Claim 11 (Original): The method of claim 3, wherein the extensible markup language is XML.

Claim 12 (Original): The method of claim 3, wherein the network router includes a packet routing engine that specifies one or more routes for data packets, and a packet forwarding engine that forwards the data packets according to the specified routes, and the method is performed in part as a process within an operating environment executing on the packet routing engine.

Claim 13 (Original): The method of claim 3, further comprising:

receiving from the client, following the replacement of the CLI process, configuration requests and operational requests encoded with extensible markup language tags;

accessing a schema that maps the tags to configuration and operational information associated with a chassis module, a device configuration module, and a routing protocol module running on a network router;

accessing the information associated with the software modules; and

emitting replies encoded with extensible markup language tags according to the schema.

Claim 14 (Currently Amended): A computer-readable medium carrying instructions to cause a programmable processor to:

establish a secure connection between a network router and a client;
initiate a command line interface (CLI) process on the network router;
receive from the client a CLI command; and
in response to the CLI command, accept commands encoded in accordance with an extensible markup language by replacing the CLI process with a management server process in response to the CLI command, wherein the management server process provides an extensible markup language-based application programming interface (API) to the client.

Claim 15 (Canceled).

Claim 16 (Previously Presented): The computer-readable medium of claim 14, wherein the instructions cause the processor to:

receive from the client configuration requests and operational requests encoded with extensible markup language tags;
access a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router, the software modules including a chassis software module that defines an inventory of components in the network router chassis, a device configuration software module that defines a physical configuration of the network router, and a routing protocol module that administers protocols supported by the network server;
parse the configuration requests and the operational requests;
access the corresponding configuration and operational information associated with the chassis software module, the device configuration software module, and the routing control software module according to the network management interface schema; and
emit replies encoded with extensible markup language tags according to the network management interface schema.

Claim 17 (Original): The computer-readable medium of claim 16, wherein the tags include chassis tags that define output for the chassis software module, device configuration tags that define output for the device configuration software module, and routing protocol tags that define output for the routing protocol software module.

Claim 18 (Original): The computer-readable medium of claim 16, wherein at least some of the replies encoded with extensible markup language tags from one or more of the software modules.

Claim 19 (Original): The computer-readable medium of claim 16, wherein the instructions cause the processor to encode at least some of the replies with extensible markup language tags based on non-encoded replies received from one or more of the software modules.

Claim 20 (Original): The computer-readable medium of claim 16, wherein the non-encoded replies include ASCII replies, the method further comprising encoding the ASCII replies in extensible markup language tags to form the encoded replies.

Claim 21 (Original): The computer-readable medium of claim 16, wherein the extensible markup language tags are defined by one or more data type definition (DTD) files.

Claim 22 (Original): The computer-readable medium of claim 16, wherein the extensible markup language tags are defined by one or more XML Schema Language files.

Claim 23 (Original): The computer-readable medium of claim 16, wherein the configuration requests include requests to change an existing configuration of the network router, wherein the instructions cause the processor to interact with one or more of the software modules to effect the requested change.

Claim 24 (Original): The computer-readable medium of claim 16, wherein the extensible markup language is XML.

Claim 25 (Original): The computer-readable medium of claim 16, wherein the network router includes a packet routing engine that specifies one or more routes for data packets, and a packet forwarding engine that forwards the data packets according to the specified routes, and wherein the instructions cause the processor to execute a process within an operating environment executing on the packet routing engine.

Claim 26 (Original): The computer-readable medium of claim 14, further comprising:

- receiving from the client, following the replacement of the CLI process, configuration requests and operational requests encoded with extensible markup language tags;
- accessing a schema that maps the tags to configuration and operational information associated with a chassis module, a device configuration module, and a routing protocol module running on a network router;
- accessing the information associated with the software modules; and
- emitting replies encoded with extensible markup language tags according to the schema.

Claim 27 (Currently Amended): A network router management interface comprising:

- a secure protocol module that provides a secure connection between a network router and a client;
- a command line interface (CLI) module that receives CLI commands from a client; and
- a management server module that receives the CLI commands from the CLI module and, in response to one of the CLI commands, accepts commands encoded in accordance with an extensible markup language,

wherein the management server module replaces the CLI module with an extensible markup language-based application programming interface (API) to the client.

Claim 28 (Canceled).

Claim 29 (Previously Presented): The network router management interface of claim 27, wherein the management server module:

receives from the client configuration requests and operational requests encoded with extensible markup language tags,

accesses a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router, the software modules including a chassis software module that defines an inventory of components in the network router chassis, a device configuration software module that defines a physical configuration of the network router, and a routing protocol module that administers protocols supported by the network router,

parses the configuration requests and the operational requests,

accesses the corresponding configuration and operational information associated with the chassis software module, the device configuration software module, and the routing control software module according to the network management interface schema, and

emits replies encoded with extensible markup language tags according to the network management interface schema.

Claim 30 (Original): The network router management interface of claim 29, wherein the tags include chassis tags that define output for the chassis software module, device configuration tags that define output for the device configuration software module, and routing protocol tags that define output for the routing protocol software module.

Claim 31 (Original): The network router management interface of claim 30, wherein the management server module receives at least some of the replies encoded with extensible markup language tags from one or more of the software modules.

Claim 32 (Original): The network router management interface of claim 29, wherein the management server module encodes at least some of the replies with extensible markup language tags based on non-encoded replies received from one or more of the software modules.

Claim 33 (Original): The network router management interface of claim 32, wherein the non-encoded replies include ASCII replies, and the management server module encodes the ASCII replies in extensible markup language tags to form the encoded replies.

Claim 34 (Original): The network router management interface of claim 29, wherein the extensible markup language tags are defined by one or more data type definition (DTD) files.

Claim 35 (Original): The network router management interface of claim 29, wherein the extensible markup language tags are defined by one or more XML Schema Language files.

Claim 36 (Original): The network router management interface of claim 29, wherein the configuration requests include requests to change an existing configuration of the network router, the management server module interacting with one or more of the software modules to effect the requested change.

Claim 37 (Original): The network router management interface of claim 29, wherein the extensible markup language is XML.

Claim 38 (Original): The network router management interface of claim 29, wherein the network router includes a packet routing engine that specifies one or more routes for data packets, and a packet forwarding engine that forwards the data packets according to the specified routes, and the management server module executes as a process within an operating environment on the packet routing engine.

Claim 39 (Previously Presented): A method comprising:

- establishing a secure connection between a network router and a client;
- initiating a command line interface (CLI) process on the network router;
- receiving from the client a CLI command;
- in response to the CLI command, providing an application programming interface (API) to receive configuration requests and operational requests encoded with extensible markup language tags;
- accessing a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router;
- parsing the configuration requests and the operational requests;
- accessing the corresponding configuration and operational information associated with the software modules according to the network management interface schema; and
- emitting replies encoded with extensible markup language tags according to the network management interface schema.

Claim 40 (Previously Presented): The method of claim 39, wherein providing an API comprises replacing the CLI process with a management server process that provides an extensible markup language-based API to the client.

Claim 41 (Previously Presented): A network router comprising:

a secure protocol module that provides a secure connection between a network router and a client;

a command line interface (CLI) module that receives CLI commands from a client; and

a management server module that, wherein, in response to one of the CLI commands, the management server module:

receives from the client configuration requests and operational requests encoded with extensible markup language tags,

accesses a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router,

parses the configuration requests and the operational requests,

accesses the corresponding configuration and operational information associated with the software modules according to the network management interface schema, and

emits replies encoded with extensible markup language tags according to the network management interface schema.

Claim 42 (Previously Presented): The network router of claim 41, wherein the management server module replaces the CLI module with the management server module and provides an extensible markup language-based application programming interface (API) to the client.